TOUCH FREE CAMERA
MENTAL COMMANDS AND HAND GESTURES

BY BENEDICT LEUNG

Undergraduate Honours Thesis

Faculty of Science
Ontario Tech University

April 2022

Benedict Leung: *Touch Free Camera*
Mental Commands and Hand Gestures, © April 2022

SUPERVISORS:
Dr. Christopher Collins

LOCATION:
Oshawa, Ontario, Canada

TIME FRAME:
April 2022

ABSTRACT

Technology has altered the immersion of our lives for the worse. As technology becomes more integrated into our lives, we spend more time with screens and lose more time in our life experiences. Using touch-free technology can lessen the divide between technology and reality and bring us closer to the immersion we once had before. In this research, hand gestures and mental commands were explored to enable interaction with a camera without holding or touching it. Hand gestures were used to change different camera modes and mental commands to initiate/execute a mode. Different camera modes are difficult to implement without the use of eye-tracking. For example, visual search relies on an object selecting a region in the scene by touching the touchscreen on your phone; using eye-tracking instead, the fixation point is used to select the region. Using multiple touch-less gestures creates more fluent transitions between our life experiences and technology.

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

1.1 MOTIVATION AND PROBLEM STATEMENT

Technology has evolved over many years and is becoming more convenient to the point of constant presence. Each evolution has changed how we live our day-to-day lifestyles. Our communication has become instantaneous and can reach anyone around the world. Our expectations on how we perceive time, distance, and relationships have changed from the past [11]. Previously, we would need to wait until the person was readily available by telephone or in-person; our expectation is disconnectivity. Now, we are expected to be connected not just in one way but in numerous ways on multiple social media platforms [11].

Social media has impacted the transition between experiences. Our life experiences have been fragmented whenever there is a downtime in our lives or when technology is presented [4]. For example, many people have been to concerts to see their favourite artists perform live on stage, which is a special in-person event. Unfortunately, many people would also like to record the moment on their phones, shattering the immersion of the present moment. Even a simple picture or internet search could ruin the moment as the process of reaching for your phone would break the immersion of the experience.

1.2 OVERVIEW

Cameras are one of the most used technologies on a day-to-day basis. It is used in various ways: social media, visual searches, messages, etc. Smoothening the transition between our life experiences and cameras, when cameras are needed to connect with people worldwide, would require another method to interact with the camera.

The focus of this thesis is on introducing another way to use a camera, making it less obstructive and interruptive to create a more immersive and fluent experience. Reaching and unlocking your phone could distract you from the life experience as it would divert your attention away from the current moment with the phone's display of unrelated notifications. The solution presented is to remove the display entirely to hide any distractions and mount the camera where the user does not need to reach for it.

Mounting the camera alone does not solve the user reaching for the device. Today's interaction involves touching the display or camera to take pictures or change device settings. Touch-less gestures were used to interact with the camera minimizing the user touching the device. Two main camera interactions were replaced with touch-less gestures: mode changing and mode execution.

Cameras not only take simple photos but can be used as a utility tool. For example, identifying and analyzing products, shopping, QR codes, videos, panoramas etc. Therefore, mode changing would require different gestures to indicate which mode is needed and require flexible gestures for adding new modes in the future. One option is using the camera to detect any hands and check the form of the hand. Using hand gestures with the camera does not need additional hardware to carry or wear, making it perfect for interacting with the camera.

The one interaction that will stay the same throughout every use case is mode execution and is also the most common interaction. Reusing hand gestures is impossible as performing the gestures will block the camera's field of view. Another considered option that requires minimal training and no hardware is blinking. However, blinking occurs unconsciously and frequently, leading to false positives and can cause uncomfortable experiences, especially if mode execution is performed in succession. Since using the camera to detect mode execution is not possible, another device needs to be added to the camera.

The optimal method when adding another device is to integrate it with the mounted camera rather than wearing two separate devices. In addition, the gesture needs to interact with the additional device without touching it but does not need to be flexible like hand gestures since initiation does not vary between each mode. The device chosen is a brain-computer interface (BCI) which measures brain activity and outputs digital signals that can be used for various interactions [3]. Although it is a separate device, future technology can integrate a camera with a BCI. The BCI also restricts the camera to be mounted on the head to measure the brain waves, but this is not a problem since the user's head is naturally looking in the direction of interest. In addition, maneuvering the device with the head involves more gross motor skills than hands, making it easier to maneuver with our heads.

One of the features of a BCI is recognizing patterns in our brain activity to create a mental command based on the pattern [3]. Although it requires training to create and execute a mental command, it is one of the only gestures that does not require physical contact with the device to interact with the device, allowing the user to be more immersive to the present moment.

Once the mode has been executed, it is essential to follow up with feedback. Since feedback needs to be sent without using a display, two other options were considered: audio and haptic feedback. Audio feedback allows the system to ask users questions about the mode (e.g. find on the internet, post on social media, etc.) or describe further details about the photo taken. Haptic feedback is simple and easy to understand but does not allow any follow-up from the user or flexibility like audio feedback does. Overall, audio feedback can make more complex interactions with the user than haptic feedback and thus is used for the feedback system.

In summary, creating the touch-free camera may allow people to be more immersed in the present moment. Touch-less gestures focus the user on what is happening rather than maneuvering the camera or smartphone.

2 RELATED WORK

This section will discuss works done in the past that are closely related to this work including hand gestures, hybrid BCIs, and search and select tasks.

2.1 HAND GESTURES

Hand gestures have been used to interact with software applications [1, 9]. Alkemade, Verbeek and Lukosch [1] used hand gestures to select what tools the user wants from the CAD software in a virtual environment. They have also explored in choosing natural gestures to relate with the conceptual design, so that it establishes a useful set of gestures to improve efficiency of the gesture-based interface [1]. Hand gestures are very flexible and can be distinct from each other with little changes to our hands. For example, the number of fingers held up and orientation of the hand [9]. This relates to the work on how gestures were chosen and how they can interact with the camera.

2.2 HYBRID BCIS

Selecting different objects with just our eyes alone can be difficult, as unintentional fixations and arbitrary dwell times can occur when users are engaged in another activity, also know as the Midas Touch problem [12]. To remove these false positives a BCI was used to replace dwell times with a mental command. There are three different types of BCIs [12]:

Active BCI: Derives its outputs from brain activity that is directly consciously controlled by the user, independently from external events, for controlling an application.

Reactive BCI: Derives its outputs from brain activity arising in reaction to external stimulation, which is indirectly modulated by the user for controlling an application.

Passive BCI: Derives its outputs from arbitrary brain activity without the purpose of voluntary control, for enriching an HCI with implicit information.

Active BCIs was used in this work, as well combining eye-tracking as the second input modality creating a hybrid BCI [12].

2.3 SEARCH AND SELECT

The effectiveness of the hybrid BCI has been evaluated by comparing past methods of search and selecting methods [10, 12]. Giving voluntary control when the selection happens gives more accurate selections [12]. Although hybrid BCIs performed slower than standalone eye-tracking devices [12], hybrid BCIs outperformed in terms of user-friendliness, and more users achieved reliable control than pure eye-tracking [10]. This work uses search and select methods proposed by hybrid BCIs to ensure the users have reliable control while maintaining user-friendliness and touch-free interactions with the device.

3 METHODOLOGY

This section will talk about the methods used to obtain a fluent experience and how they can smoothen the transition between our life experiences and when reaching for our camera. This section will also describe why and how these methods interact with the camera. All the chosen methods are touchless, meaning the user does not need to touch any devices to interact after the device has been worn.

3.1 MENTAL COMMANDS

Mental commands are effective in minimal physical contact with the device. Reducing physical contact can increase the immersion of the experience by allowing the user to be more immersive rather than maneuvering the device with their hands. Mental commands depend on recognizing patterns in the user's brain activity and learning the difference between the user's neutral state and the desired command state [3]. So, it is optimal to choose a thought that can be distinct and as strong as possible in a neutral state. The thought can be literal or abstract (e.g. the mental command "lift" can associate with lifting a virtual box (literal) or visualizing a scene (abstract)). The choice of method depends on the user's disposition towards a certain modality [3]. A strong disposition towards a certain modality of sensory input (touch, sound, taste, etc.) can focus on the command state easier [3]. Multiple command states are possible but avoided since it requires intensive training to be comfortable with two or more commands and trigger them independently. Therefore, using one mental command must be associated with a constant interaction throughout all the camera use cases. The interaction used to associate with the command is mode execution.

3.2 HAND GESTURES

As mentioned, only one command state is used for interacting with the camera in all the use cases, but the purpose of a photo can vary between each use case. So, it is required to have a dynamic gesture that can account for future and present purposes. Our hands can be changed in various ways, for example, the position and curvature of our fingers. This feature makes hand gestures a good candidate for handling different use cases as it could associate one hand gesture to one or more use cases. This gesture also does not require any additional hardware to recognize since images of hands are sufficient and accurate enough to trigger each hand gesture independently. Different camera modes are implemented to fit the user's needs or requirements. So, one hand gesture will be associated with one camera mode.

3.3 EYE TRACKING

Images can be very useful in many ways, but images themselves provide little to no context to what the user is focusing on in the scene. The only way to determine that context is if the scene only contains one object or is more significant than all the other objects. One possible solution is eye-tracking to determine where the user is interested in the scene. Eye-tracking allows different applications to be possible that images alone cannot provide. Combining eye tracking with image processing algorithms can provide more context and detail, making different camera modes possible without needing a display or touching any devices.

3.4 FEEDBACK

Achieving the most immersive experience would require the user not to look at any display. Although a display is very effective in showing results and progress, the program must send feedback without any display. A possible alternative to let users know what is going on with the progress is audio feedback. Audio feedback can have many forms, such as audio cues (sound effects) and speech. Audio cues can convey different kinds of information in a short time. For example, one sound effect can associate when an action is starting, progressing, or finished, making the user understand what the device is doing at all times. Associating sound effects with a status needs to relate to (e.g. shutter sound when the picture is taken); otherwise, it will be difficult to understand what information is trying to be conveyed. Speech is used for conveying complex information that audio cues cannot do understandably in a short time. In a photo-taking context, speech can explain details about a photo such as objects seen, location, visual sentiment, etc. The context can be more specific to the user's interest and provide more meaningful feedback about the photo using eye-tracking.

4 IMPLEMENTATION

This section will discuss the hardware used to detect gestures described in the previous section. It will introduce what APIs in the software architecture is used to recognize gestures and how they are recognized. It will also introduce what camera modes are implemented to showcase how the methods interact.

4.1 HARDWARE

Our classic camera device needs to be modified to recognize mental commands and hand gestures. Also, the camera should not be maneuvered by our hands to achieve maximum immersion. All the devices used are, as a result, are worn on the head. Maneuvering a camera with a head requires fewer motor skills than maneuvering with hands. Since the head area involves larger muscle groups (gross motor skills) than our hands (fine motor skills). Therefore, it requires less skill to maneuver a camera with our head than our hands, and also, it is intuitive to look at a scene to take a photo of it.

4.1.1 *Brain Computer Interface*

Brain-computer interfaces acquire the wearer's brain signals and analyzes them to execute the desired action. EMOTIV Insight (Figure 1) was used as the brain-computer interface. It provides five-channel EEG (AF3, AF4, T7, T8, Pz), which EMOTIV uses to create four data streams (mental commands, performance metrics, facial expressions, and motion sensors) [3]. EMOTIV Insight uses Bluetooth 5.0 to wirelessly connect to devices, which means no hassle with wires and minimal time to set up.

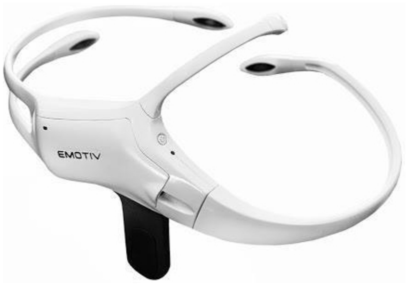


Figure 1: EMOTIV Insight provides 5 channel EEG that allows mental command creation

4.1.2 *Eye Tracking Glasses*

Pupil Core (Figure 2) and Pupil Capture software [2] was used for eye tracking. The glasses have two cameras: a world camera and an eye camera. The world camera captures whatever the user is facing with a FOV of 99° x 53° and acts as the user's camera. The eye camera captures the eye to estimate gaze and fixation points with an accuracy of 0.60° after a 5 point calibration. The device is connected via USB to connect to a mobile device or personal device to send data into the network. The software is continually listening at one of the ports.

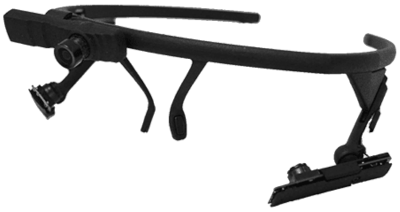


Figure 2: Pupil Core provides eye-tracking data that can estimate gaze/fixation points over the network. It also captures images from the world at thirty frames per second at 720p.

4.1.3 *Wireless Earbuds*

Since audio feedback is used, it would be optimal for the people around the user not to hear anything from the device. Wireless earbuds are used to hear any feedback from the camera with minimal time to set up and no wires in the way of the user. One earbud is worn as both earbuds could distract the user from life experiences, but still can switch attention for a brief moment to listen for any updates or audio cues that confirm the user's needs and requirements.

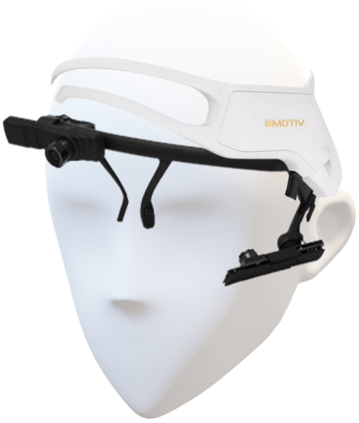


Figure 3: Users will wear all the devices mentioned above at the same time.

4.2 ARCHITECTURE

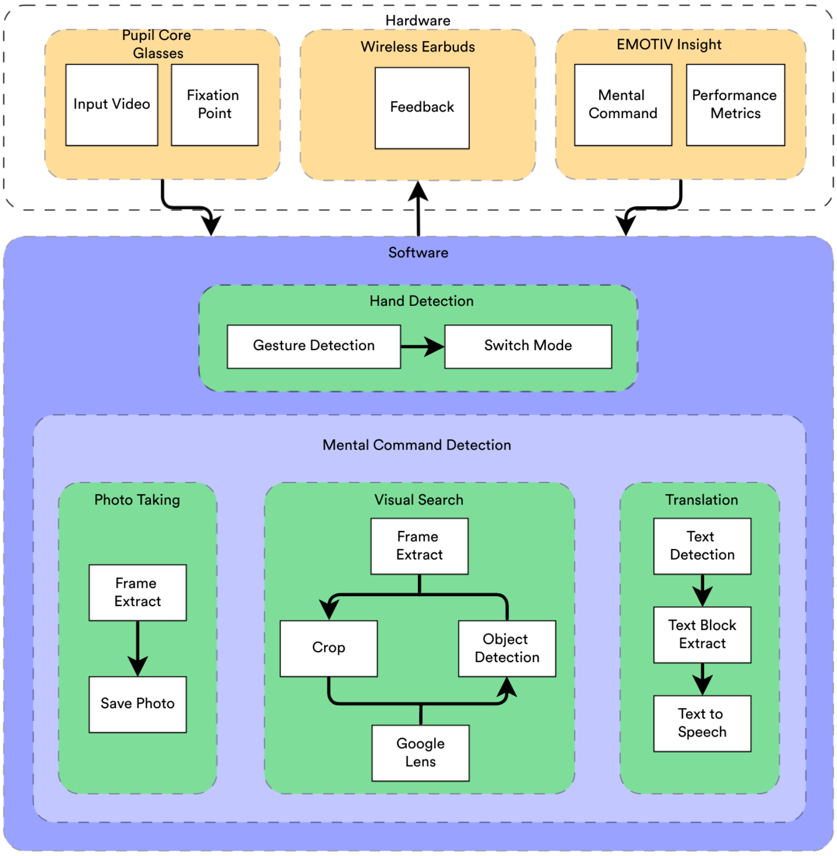


Figure 4: A diagram to represent the software architecture. Each colored node in the software node represent a thread and each white nodes represent action taken place

The software architecture (Figure 4) uses multi-threading in Python to listen for data that the hardware is sending over the network. Two threads are deployed to handle hand recognition and mental commands from the start. The hand recognition thread processes the model of the hand to determine which gesture is being performed and results in a mode change. The mental command thread processes mental command data to determine if the user wants to execute a mode. It then spawns a thread responsible for processing the image based on the mode. Each thread has been laced with feedback to ensure the user understands what is happening at all times. Multi-threading allows the system to fetch and process data concurrently to ensure all information produced is as close to real-time as possible.

4.3 HAND DETECTION

Hand detection is required to detect hand gestures. MediaPipe Hand [8] utilizes a machine learning model to infer twenty-one 3D landmarks of the hand (Figure 5) on a single frame and performs real-time detection with multiple hands on a mobile phone. This lightweight model suits detecting gestures by comparing the coordinates of these landmarks.

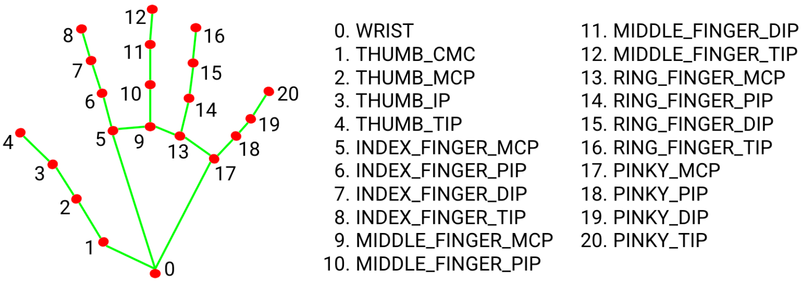


Figure 5: Twenty one landmarks of the hand provided by MediaPipe [8]

The gestures used are the number of fingers holding up, which is associated with one mode. The number of fingers is determined by comparing the fingertip coordinates (landmark 4, 8, 12, 16, and 20) to the PIP joints or the MCP joint for the thumb (landmark 2, 6, 10, 14, and 18). The finger is counted by comparing if the y-coordinate of the fingertips is higher than the y-coordinate of the PIP joints. For the thumb, the finger is protruded outwards to the side, so the x-coordinate is used instead.

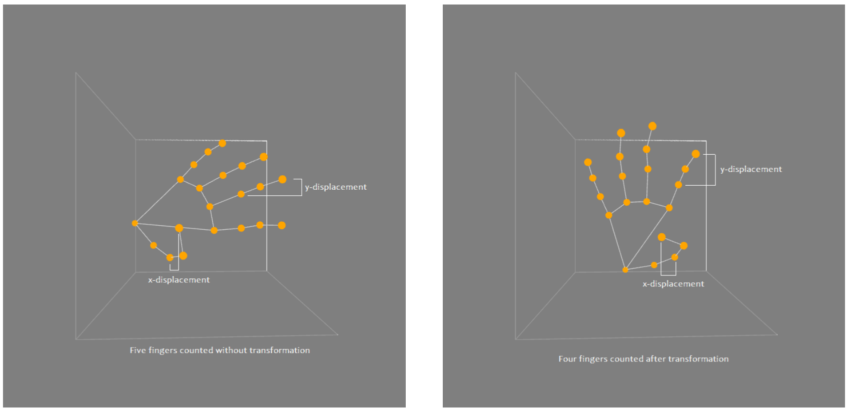
The orientation of the hand (i.e. whether the palm is facing towards you or away), the handedness (i.e. left or right) and the rotation of the palm was also considered. The coordinates were transformed based on the angle of the palm:

$$\theta = \arctan\left(\frac{landmark\_9.y - landmark\_0.y}{landmark\_9.x - landmark\_0.x}\right)$$

$$x' = x \cdot \sin(\theta) + y \cdot \cos(\theta) \text{ (for thumb)}$$

$$y' = y \cdot \sin(\theta) - x \cdot \cos(\theta) \text{ (for rest of the fingers)}$$

This simple transformation handles the rotation of the wrist to ensure the hand is upright, as shown in Figure 6. When the hand is not upright, fingers can be miscounted since the fingers do not curl in the direction of the y-axis (x-axis for the thumb). The hand's orientation is used to transform the coordinates of the thumb by comparing the transformed coordinates of landmarks 17 and 1. The orientation paired with the handedness will determine the negation of the angle for the thumb.

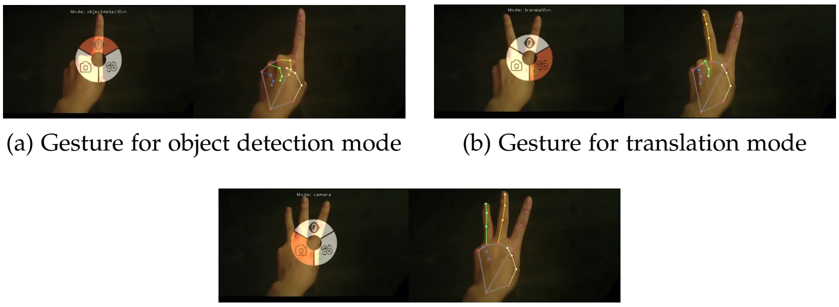

y-displacement
x-displacement
Five fingers counted without transformation
y-displacement
x-displacement
Four fingers counted after transformation

(a) Five fingers counted before transformation

(b) Four fingers counted after transformation

Figure 6: Four fingers are held up in this example. (a) Five fingers are counted before transformation, the thumb has been miscounted since the fingertip's x-coordinate is larger than the thumb's MCP joint (b) Four fingers are counted after transformation

Figure 7 shows all the gestures for each of the modes needed to perform to switch between different modes. Note that fingers can be hidden, shown in Figure 7, but MediaPipe handles this by assuming the fingers are curled. The gesture needs to be held for 2 seconds straight to change the mode. The software gives two types of feedback for mode switching: subtle progress sound for detecting a gesture during the 2 seconds and speech for the mode currently on at the end of the 2 seconds (e.g. "Camera mode," "Translation mode").



(a) Gesture for object detection mode

(b) Gesture for translation mode

(c) Gesture for camera mode

Figure 7: On the left section of each image is the gesture detected. For visual purposes, there is a pie menu indicating the mode in orange drawn with OpenCV. On the right, is the hand detection model provided by MediaPipe.

4.4 MENTAL COMMAND TRAINING

Training is required for the brain-computer interface to recognize user-specific brain activity patterns to create mental commands. The software used EmotivBCI [3] software for quality checks, training and mental command creation. To ensure the quality of the training, contact and EEG quality checks are performed by the software. All five sensors are green, and the baseline is taken before any training is performed. Two command states are trained: neutral state and the desired command state. A neutral state is used when the user is idling or the state the camera is ignoring. The command state is used for executing a mode.

The training process used is ten runs per state, twenty in total. Each run takes eight seconds to finish, and after each run, training feedback (1-100) is received. The goal specified by EmotivBCI is seventy-five. Any lower scores would result in a rejection of the run. The training sequence used was to alternate between the neutral state and the desired command state to better contrast the two states [3]. After ten successful runs, the software adjusted the sensitivity to make it easier to trigger the desired command state. On a scale of 1-10 with a default of five, the sensitivity used is seven for the desired command state and five for the neutral state.

Mental commands are used to initiate the mode the user is currently on. To avoid false positives on mental command detection, the user must execute the command for two seconds straight without interruptions to initiate the mode and a power level above 65 provided by EmotivBCI. This process uses a shutter sound effect as the feedback at the end of the two seconds.

4.5 MODES

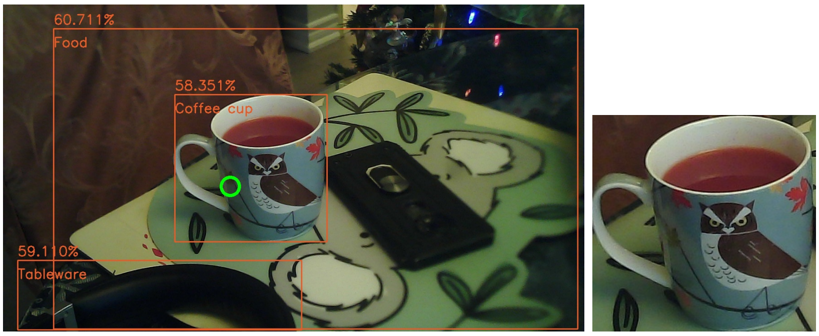
Photo taking, object detection, and translation are the implemented modes. These modes are chosen based on the possible scenarios that touch-less gestures would be most effective. Photo taking is a simple model that captures a scene. This mode acts as the camera's main feature that creates a photos gallery. Object detection mode combines eye-tracking and objects localization algorithms to provide more information about the user's object. Finally, translation mode translates from whatever the language is detected to English using optical character recognition and translation API. These modes present a solution to cover all the use cases for a camera by executing a hand gesture associated with the desired mode.

4.6 translation

Translation has been used for many purposes, and it could help with academics or help with navigating a foreign country. When being in an environment with a foreign language, translation will be needed multiple times in succession. Also, without using a touch display to specify what text to translate, eye tracking will be used to extract a block of text from Google Vision's optical character recognition (OCR) [7]. Text extraction can occur whenever the fixation point changes (i.e. fixation duration of 400ms) and when the previous translation's audio feedback has finished playing to minimize the user executing the mode in succession. The extracted text will then be translated into English with Google Translation [6] and spoken back to the user. This mode uses the mental command to toggle on/off the translation.

4.7 object detection

Visual search is designed to search for information on the Internet using images. Inputting images to the visual search engine could be difficult since noise could be presented in the image, resulting in inaccurate results. For example, multiple objects are present or the background of the image is complex. A solution presented is to combine eye-tracking and object localization algorithms to crop the noise out of the image.



(a) Bounding boxes for the full image

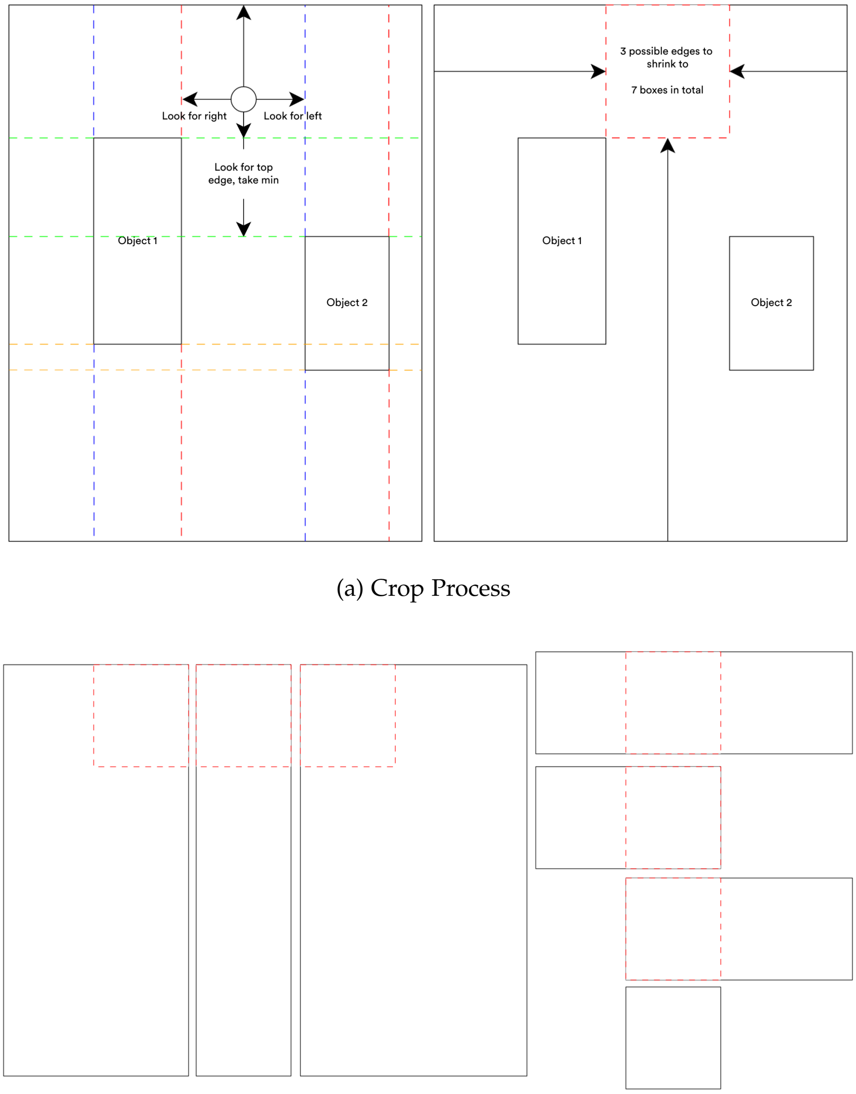
(b) Object crop (Coffee Cup)

Figure 8: (a) A frame extract after mental command execution. Orange boxes indicate bounding boxes of the objects detected by Google Vision API [7] with labels and confidence levels. Green circle indicates the fixation point. (b) The object detected determined by the fixation point.

Figure 8 shows the process of the object detection mode. The process utilizes Google's object localization algorithm, Google Vision API [7], and a custom cropping algorithm to remove noise from the image. Using bounding boxes from Google Vision API, the software can isolate the object into a visual search engine without any objects affecting the final result.

4.7.1 *Improving Google's Object Localization Algorithm*

Using only Google Vision is not sufficient to detect all objects in the scene, as seen in Figure 8. In that example, the mobile phone beside the coffee cup has been left out and not detected. In addition, there are bounding boxes that can be very abstract and encapsulate multiple objects at a time. Therefore, it is not sufficient to use one pass of Google Vision.



(a) Crop Process

(b) Crop Boxes

Figure 9: (a) The crop algorithm utilizes a box to produce multiple crops to crop out some or all of the objects. (b) The results of the crop algorithm.

The solution proposed is if there are no objects detected inside the fixation point, crop out the detected objects from the scene to pass to the Google Vision again. Figure 9a shows the steps of the cropping process. First, it finds the biggest box where if the edges are protruded outwards one at a time, the edges will not intersect any bounding boxes and contain the fixation point. Note that the box can have edges aligned with the edge of the image. Finally, the image shrinks its edges to the box one edge up to four edges at a time (Figure 9b). The total of crops produced by this method is:

$$c_{total} = \sum_{k=1}^{e_f} \binom{e_f}{k}$$

Where e_f is the number of edges not aligned with the image, each crop will produce another image which will be passed on to Google Vision again.

Two cases will continue to the next iteration and restart the process: (1) The existing bounding boxes do not contain the fixation point which will initiate the custom crop algorithm (left crop in Figure 10) (2) More than one object has been detected, and at least one of its bounding box must contain the fixation point which will be cropped out of the image instead of using the custom crop algorithm. The process ends if only one object has been detected and its bounding box contains the fixation point (middle crop in Figure 10) or no object has been detected (right crop in Figure 10). In this example, the first pass has not detected the mobile phone, but cropping the image into subsections can help guide Google Vision and focus on what regions need more attention.

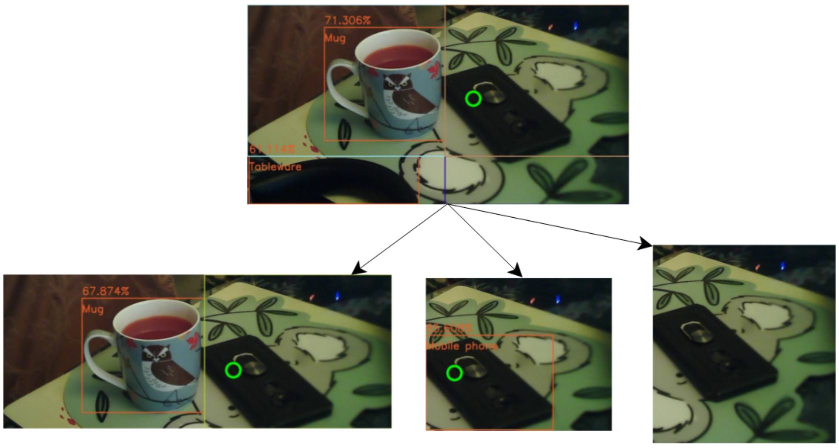
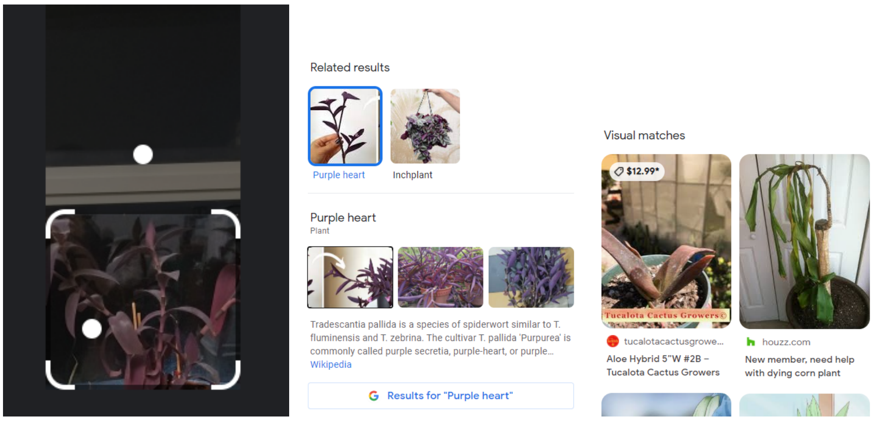


Figure 10: A demo of the crop algorithm where $e_f = 2$. This demo only shows one stage of the crop process. The left image shows how to produce the box and the right shows how it is used to produce crops.

4.7.2 *Better Labels*

The labels provided by Google's object localization algorithm are not very meaningful. Using object detection to obtain more specific labels can be effective after the noise has been reduced. Using a web driver for Google Chrome (Chrome version 98), the upload process to Google Lens [5] is automated to scrape results from the web page.



Tradescantia pallida is a species of spiderwort similar to T. fluminensis and T. zebrina. The cultivar T. pallida 'Purpurea' is commonly called purple secretia, purple-heart, or purple...

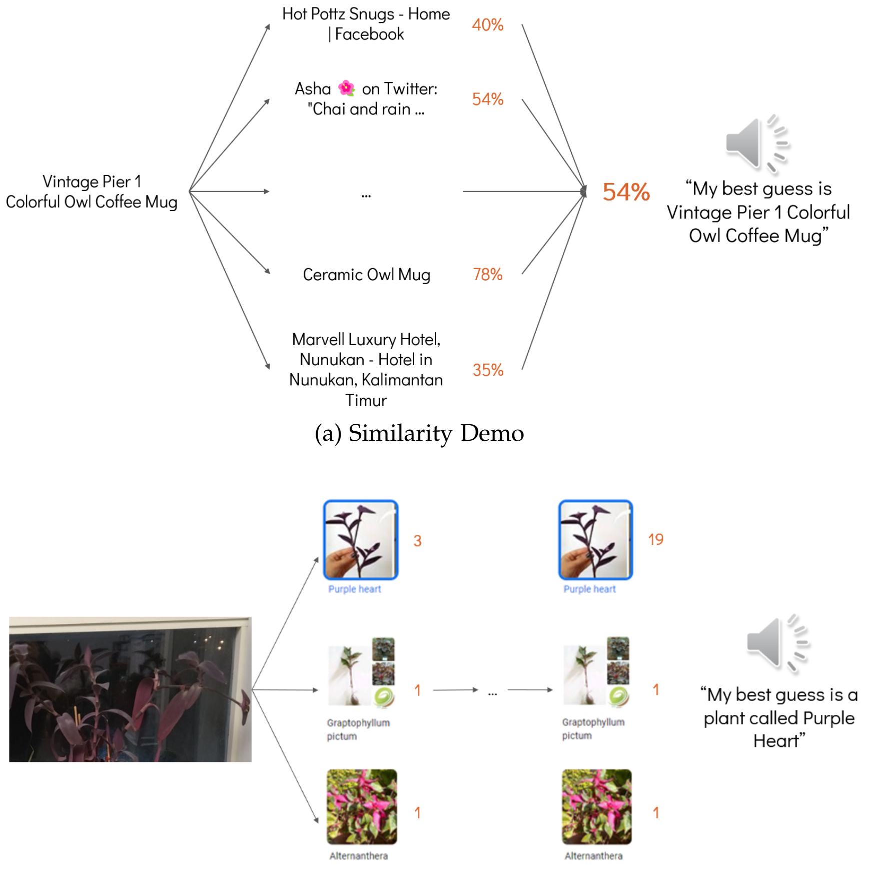
(a) Region Selector (b) List of labels (c) Web pages

Figure 11: (a) Google Lens are able to detect visual matches from a sub-region indicated by white circles. The border indicates the sub-region Google Lens is currently on (b) Multiple labels can be presented with the displayed object being the most confident (c) No labels are found, so the title of web pages are extracted instead.

Since the cropping algorithm produces many images from the original image, a point system is used to count the occurrences the labels appear in each of the images provided by Google Lens [5]. First, the web driver will look for if there are any detections made by Google Lens, shown in Figure 11a, and will select the one closest to the fixation point. Two different results are possible: (1) Google Lens will provide a label (i.e. Spotted eagle-owl, Purple plant, SpongeBob SquarePants, etc.) (Figure 11b) (2) Google Lens will provide any web pages related to the object (Figure 11c). In some cases, Google Lens can provide a list of labels and display the one that is the most confident (Figure 11b). The most confident label is assigned for three points, and the rest of the list is assigned one point. If Google Lens cannot provide any labels, then the web driver will look for web pages instead. Two lists have been created by the end of the scraping process: labels and web page titles.

The algorithm will choose the label with the most points for the user. If there are any ties between the labels, web page titles will be checked for these labels and updated in the point system for every occurrence. If the tie remains, pick one out of the tied labels randomly. The algorithm will instead compare the web page titles if no labels are found. Each title will check its similarity with other titles using spaCy [13], a natural language processing library. spaCy converts sentences into vectors using embedding and can be compared to check the similarity. As a result, each web page title will have a percentage score (0%-100%) for each other titles. The arithmetic mean (average) is then calculated to represent the average similarity of all the other titles. The algorithm will choose the title with the highest average similarity.

Once the label has been extracted. The label is spoken to the user using text to speech. Figure 12 shows a demo of how data is extracted and processed from Google Lens. In summary, web page titles will be compared when no labels are found across all of the images produced by the cropping algorithm and will use the web page title with the highest average similarity as the label for the object (Figure 12a). When labels are present, the point system is used to count the occurrences across all of the images and will take the maximum in the point system (Figure 12b).


Vintage Pier 1 Colorful Owl Coffee Mug
Hot Pottz Snugs - Home | Facebook
40%
Asha 🌺 on Twitter: "Chai and rain ...
54%
...
Ceramic Owl Mug
78%
Marvell Luxury Hotel, Nunukan - Hotel in Nunukan, Kalimantan Timur
35%
54%
"My best guess is Vintage Pier 1 Colorful Owl Coffee Mug"
Purple heart
3
Graptophyllum pictum
1
Alternanthera
1
...
Purple heart
19
Graptophyllum pictum
1
Alternanthera
1
"My best guess is a plant called Purple Heart"

(a) Similarity Demo

(b) Point System Demo

Figure 12: (a) When no labels are found, each web page title will be compared with the other web page titles and obtain the average similarity. (b) When labels are found, the first label is assigned 3 points and the rest of the labels 1 point.

5 CONCLUSION

This section will talk about the work accomplished and its limitations. It will also describe the future of this work, including potential, evaluation and ethical considerations.

5.1 LIMITATIONS

Touch-free gestures with a head mounted device can help users focus on the present moment rather than looking at a display and maneuvering the device. Wearing a brain-computer interface that wraps around the head can be uncomfortable to wear for long periods and is sensitive to movement that can hinder the quality of EEGs. In addition, training for mental commands can be difficult for new users who have not experienced using the BCI. Intentionally changing our brain activity can vary differently between users, so it is difficult to explain how it can be done. Furthermore, brain activity changes rely on our experiences, and it is impossible to explain the experience to another person, the same problem that arises in explaining colours. Finally, a calibration process is required before use to check for the quality of the EEGs, which is not practical for everyday use.

In terms of the eye-tracking glasses, the camera's field of view does not cover the field of view of our eyes. So, the user's head needs to be adjusted accordingly to fit the object of interest in the camera frame properly, or the user's hand needs to be at a distance away before the hand can be in the camera frame. Like the BCI, the eye-tracking glasses used for this work need a 5-point calibration every time it is put on to reduce the errors in the location of the fixation points.

Since both devices are worn at the head, they can be hinder each other's performance as the eye-tracking glasses need to be worn underneath the BCI, which results in lower contact quality. On the other hand, the BCI also needs to be readjusted to ensure optimal EEG quality, moving the eye-tracking glasses and offsetting the fixation point.

All hardware limitations can be solved in the future. A BCI integrated with eye-tracking glasses can solve both devices hindering each other's performance. Smaller BCIs could be more comfortable to wear for long periods, and some technologies do not require a calibration process for eye-tracking, which results in faster setup.

On the software side, the automation of the upload process to Google Lens relies on a web driver and scrapes data on the website. Websites can be easily changed, making the automation fail—for example, different HTML structures and class names. Furthermore, Google Vision API uses a network to send and receive data which can slow down the process of the camera modes. These can be fixed with local APIs, which can eliminate network latency, but it is unsure if Google Lens will develop an API to use in the future.

5.2 FUTURE WORK

Many optimizations can be made to the touch-less camera but are not implemented due to accessibility of the technology and time constraints. Also, the methods used in this work can be easily be changed, such as different gestures or different hardware.

5.2.1 *Hardware Optimizations*

Today's market has a smaller BCI that can be worn on the back of the head, making the BCI be integrated to have eye-tracking and allow attachment to the eye-tracking glasses' temple tips. Furthermore, state-of-the-art eye-tracking glasses do not need a calibration process and look similar to eyeglasses with exchangeable lenses. Therefore, glasses do not need to be worn with the device. These optimizations can allow the device to be taken off more easily and reduce the time to set up and calibrate while being more comfortable to wear.

5.2.2 *Gestures*

The potential of this work can be extended with different interactions, such as voice recognition and haptic. Using voice recognition can improve false positives from the BCI. The prevention of false positives used for BCI is executing a mental command for two seconds and a power level above 65. Executing a mode would take two seconds after the initial intention. Using voice recognition could eliminate this. Wearable haptic technology could also replace hand gestures to avoid other hands being detected and be more resistant to false positives with a downside of a separate device that needs to be worn.

5.2.3 *Predict Modes*

Hand gestures were used to change different modes to fit the intended use case, but eye-tracking can predict this. Translation mode will be predicted if a foreign text is detected at the fixation point. The software can also ask yes or no questions to confirm the intended mode further. For example, "Do you want to translate?" If a mental command has been detected, it will execute the mode. Furthermore, asking questions can combine and extend different camera modes, such as object detection, into buying the product online.

5.3 EVALUATION

Due to time constraints, the work will not be evaluated with participants, but the proposed evaluation process is as follows. First, participants will undergo two phases of mental command training. The first phase will let the participants explore how EmotivBCI works, familiarize themselves with creating a mental command, and determine what is possible to detect changes in their brain activity. The second phase will undergo the training for two mental commands (neutral state and desired command state) that are specified in Section 4.4.

Several objects will be placed in a free space room with colour-coded stickers indicating which mode they will be used. Participants will be asked to carry out all queries, in any order. Giving free space to participants will give an experience close to a real-world application rather than giving instructions or paths to take when using the device. This method will reveal if there are problems with the device's usability and difficulty.

Since the evaluation process does not follow a single path, data needs to be recorded for later observations to ensure the touch-free camera is used and works as intended. The room needs to be recorded to observe how users interact with the object and the touch-free camera. Next, data from the hardware also needs to be recorded to ensure that the hardware responds to the user's intentions. Visual representations of data have already been implemented (Figure 7). The data that has been shown in the figure includes fixation points, camera mode, hand model, and mental command duration, which are shown on top of the camera feed. Finally, results from executing a mode need to be recorded to ensure the algorithms and APIs are interacting with each other as intended. Results include bounding boxes, current fixation point and audio feedback. A questionnaire will be given to collect participant feedback on performance and usability at the end.

5.4 ETHICAL CONSIDERATIONS

There are ethical concerns about how this work allows users to take pictures without noticing potentially invading someone's privacy. Furthermore, looking at someone's brain activity to predict or determine what the user is doing could also invade their privacy. Although current technology only allows understanding of general cognitive processes and not full semantics of thoughts, it could predict if the user is processing language or engaged in some activity. Combining with another stimulus could make more possible, so storing BCI data should be carefully considered as it could capture the user's neural signals and, as a result, could replicate the mental command.

No implementation or action was taken to solve these ethical concerns. However, it is acknowledged that it is an issue that must be addressed if this technology ever goes out to the public.

5.5 CONCLUSION

This work presents solutions to create a fluent experience when using a camera. It explores combining a brain-computer interface and eye-tracking glasses to create a touchless camera and take pictures with thoughts. Using fixation points, the user can select a frame region for the camera to act on. Overall, the hardware used replaces touch interactions with touch-less ones, allowing the user to be more attentive to the present moment.

BIBLIOGRAPHY

[1] Remi Alkemade, Fons J. Verbeek, and Stephan G. Lukosch. "On the efficiency of a VR hand gesture-based interface for 3D object manipulations in conceptual design." In: *International Journal of Human–Computer Interaction* 33.11 (2017), pp. 882–901. DOI: 10.1080/10447318.2017.1296074. eprint: https://doi.org/10.1080/10447318.2017.1296074. URL: https://doi.org/10.1080/10447318.2017.1296074.

[2] Moritz Kassner, William Patera, Andreas Bulling. "Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction." In: (2014). URL: https://doi.org/10.1145/2638728.2641695.

[3] EMOTIV. *EmotivBCI*. 2019. URL: https://emotiv.gitbook.io/emotivbci/.

[4] Mary Gomes. *Five Reasons to Take a Break from Screens*. 2018. URL: https://greatergood.berkeley.edu/article/item/five_reasons_to_take_a_break_from_screens.

[5] Google. *Google Lens*. 2022. URL: https://lens.google/.

[6] Google. *Translation AI*. 2022. URL: https://cloud.google.com/translate.

[7] Google. *Vision AI*. 2022. URL: https://cloud.google.com/vision.

[8] MediaPipe. *MediaPipe*. 2020. URL: https://google.github.io/mediapipe/solutions/hands.

[9] Meenakshi Panwar. "Hand gesture based interface for aiding visually impaired." In: *2012 International Conference on Recent Advances in Computing and Software Systems*. 2012, pp. 80–85. DOI: 10.1109/RACSS.2012.6212702.

[10] Piotr Stawicki, Felix Gembler, Aya Rezeika, and Ivan Volosyak. "A novel hybrid mental spelling application based on eye tracking and SSVEP-based BCI." In: *Brain Sciences* 7.4 (2017). ISSN: 2076-3425. DOI: 10.3390/brainsci7040035. URL: https://www.mdpi.com/2076-3425/7/4/35.

[11] Jim Taylor. *Children's Immersion in Technology is "Shocking"*. 2012. URL: https://www.psychologytoday.com/ca/blog/the-power-prime/201209/children-s-immersion-in-technology-is-shocking.

[12] Thorsten O. Zander, Matti Gaertner, Christian Kothe, and Roman Vilimek. “Combining eye gaze input with a brain–computer interface for touchless human–computer interaction.” In: *International Journal of Human–Computer Interaction* 27.1 (2010), pp. 38–51. DOI: 10.1080/10447318.2011.535752. eprint: https://doi.org/10.1080/10447318.2011.535752. URL: https://doi.org/10.1080/10447318.2011.535752.

[13] spaCy. *spaCy*. 2022. URL: https://spacy.io/.